Neural Networks in the Infinite Width Limit and the Neural Tangent Kernel

Mateusz Mroczka[1] and Benedikt Petko[2]

[1] *Department of Mathematics, University of Oxford*
[2] *Department of Mathematics, Imperial College London*

Abstract

We summarize recent developments in the analysis of infinite width limits of dense neural networks with random parameter initialization. In the finite width case, the classical gradient descent on the parameter space can be interpreted as a kernel gradient descent in a suitable space of functions with respect to a kernel depending on the network architecture. This kernel is in the literature known as the neural tangent kernel. It is stochastic due to random initialization; however, it has been shown that, as the widths of hidden layers are taken to infinity, these kernels converge to a deterministic limiting kernel. Under certain assumptions, the kernel gradient descent with respect to this limiting kernel also corresponds to gradient descent on parameters in the infinite width limit. We demonstrate this correspondence in our own numerical experiments.

Contents

1 Introduction

Before motivating the study of neural networks in the infinite width limit, we will introduce the setup in which we will be working in.

1.1 Neural network model

Definition 1.1 (Dense neural network)**.** [4] Consider a neural network with $L+1$ layers indexed by $0, \dots, L$ with $n_0, \dots, n_L$ nodes respectively. Each pair of neighbouring layers has an associated connection matrix $W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}$ and bias $b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}}$ where $\ell = 0, \dots, L-1$. This gives a total of

$$P := \sum_{\ell=0}^{L-1} (n_\ell + 1) n_{\ell+1}$$

parameters in the neural network and we assume that all parameters are initialized by sampling from a standard normal distribution. In this setup, we construct a network function $f_\theta : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ given by $f_\theta(x) := \tilde{\alpha}^{(L)}(x;\theta)$, where preactivations $\tilde{\alpha}^{(\ell)}(\cdot;\theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}$ and activations $\alpha^{(\ell)}(\cdot;\theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}$ are defined recursively by

$$\begin{aligned}
\alpha^{(0)}(x;\theta) &= x \\
\tilde{\alpha}^{(\ell+1)}(x;\theta) &= \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \alpha^{(\ell)}(x;\theta) + \beta b^{(\ell)}, \quad l = 0, \dots, L-1 \\
\alpha^{(\ell)}(x;\theta) &= \sigma(\tilde{\alpha}^{(\ell)}(x;\theta)), \quad l = 1, \dots, L-1
\end{aligned}$$

In our setup, $\sigma : \mathbb{R} \to \mathbb{R}$ is a Lipschitz, almost everywhere twice differentiable function with bounded second derivative. Whenever σ is applied to a vector, we have entrywise evaluation in mind.

Definition 1.2 (The network training map)**.** [4] Let $\mathcal{F}$ be a suitable space of functions, $C : \mathcal{F} \to \mathbb{R}$ a cost functional and let $F^{(L)} : \mathbb{R}^P \to \mathcal{F}$ be the function that maps a parameter vector to a corresponding function output by an $L+1$ layer dense neural network as in the previous definition, that is

$$F^{(L)}(\theta) := \tilde{\alpha}^{(L)}(\cdot;\theta) \in \mathcal{F}.$$

Note that the activation function is not applied to the output layer. Alternatively, denote

$$f_\theta := F^{(L)}(\theta).$$

1.2 Infinitely wide networks as Gaussian processes

Given that parameters θ of the neural network are initialized as independent samples from a standard normal distribution, it has been shown in [6][5] that the network function $f(x;\theta)$ converges in law to a Gaussian process as the width of the hidden layers tends to infinity. This result is due to [6] for the case $L = 1$ and [5] for the general case as stated in the following theorem with proof in the appendix.

Theorem 1.3. [5] For any sample $\{x^1, \dots, x^N\} \subset \mathbb{R}^{n_0}$ the n_L random vectors $\big(f_{\theta,k}(x^1), ..., f_{\theta,k}(x^N)\big)$ indexed by $k = 1, ..., n_L$ tend in law, as $n_1, ..., n_{L-1} \to \infty$, to i.i.d. N-dimensional Gaussian random variables with zero mean vector and co-variance matrix $\Sigma^{(L)}$ with entries

$$\Sigma^{(L)}_{ij} = \Sigma^{(L)}(x^i, x^j), \quad i, j = 1, ..., N$$

and the kernel $\Sigma^{(L)}(\cdot,\cdot)$ defined recursively through

$$\begin{aligned}
\Sigma^{(1)}(x,x') &= \frac{1}{n_0} x^T x' + \beta^2, \\
\Sigma^{(\ell+1)}(x,x') &= \mathbb{E}_{(u,v)\sim\mathcal{N}\left(0,\Lambda^{(\ell)}(x,x')\right)} \left[\sigma(u)\sigma(v)\right] + \beta^2,
\end{aligned}$$

where

$$\Lambda^{(\ell)}(x,x') = \begin{pmatrix} \Sigma^{(\ell)}(x,x) & \Sigma^{(\ell)}(x,x') \\ \Sigma^{(\ell)}(x',x) & \Sigma^{(\ell)}(x',x') \end{pmatrix} \in \mathbb{R}^{2\times 2}.$$

1.3 Positive definite kernels

We expound the formalism introduced by Jacot et al. [4] necessary to define the neural tangent kernel.

Definition 1.4. [3] Let $\mathcal{F}$ be a space of functions from $\mathbb{R}^{n_0}$ to $\mathbb{R}^{n_L}$. A functional $C : \mathcal{F} \to \mathbb{R}$ is said to be convex if for every $f_1, f_2 \in \mathcal{F}$ and $\lambda \in [0,1]$,

$$F(\lambda f_1 + (1-\lambda)f_2) \leq \lambda F(f_1) + (1-\lambda)F(f_2).$$

We can interpret optimizing $C \circ F^{(L)} : \mathbb{R}^P \to \mathbb{R}$ as optimizing the functional $C : \mathcal{F} \to \mathbb{R}$ itself by using the notion of a derivative of a functional as defined below. In the usual settings, this functional is convex, as will be shown in the case of the linear regression cost functional.

Remark 1.5. For the following arguments, we can assume that, for example, $\mathcal{F} = C_c^\infty(\mathbb{R}^{n_0};\mathbb{R}^{n_L})$.

Definition 1.6 (Multi-dimensional kernel)**.** [4] A multi-dimensional kernel is a measurable map

$$K : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}^{n_L \times n_L}$$

such that $K(x,x') \in \mathbb{R}^{n_L \times n_L}$ is a symmetric matrix for all $x, x' \in \mathbb{R}^{n_0}$.

Definition 1.7 (Two positive semi-definite symmetric bilinear forms)**.** [4] Let $p^{in} = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$ be the empirical distribution of the data. Then the map

$$\begin{aligned} &\langle \cdot,\cdot \rangle_{p^{in}} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}, \\ &\langle f,g \rangle_{p^{in}} = \mathbb{E}_{x\sim p^{in}}[f(x)g(x)] = \frac{1}{N}\sum_{i=1}^N f(x_i)g(x_i), \end{aligned}$$

defines a positive semi-definite bilinear form on $\mathcal{F}$. Given a multidimensional kernel K, we can also define the symmetric bilinear form

$$\begin{aligned} &\langle \cdot,\cdot \rangle_K : \mathcal{F} \times \mathcal{F} \to \mathbb{R}, \\ &\langle f,g \rangle_K = \mathbb{E}_{x,x'\sim p^{in}}[f(x)^T K(x,x') g(x')]. \end{aligned}$$

A multidimensional kernel K is said to be positive definite with respect to $\|\cdot\|_{p^{in}}$ if

$$\forall f \in \mathcal{F} : \|f\|_{p^{in}} > 0 \implies \|f\|_K > 0$$

in which case the bilinear form $\langle \cdot,\cdot \rangle_K$ is positive semi-definite.

Definition 1.8 (Dual with respect to the empirical distribution)**.** [4] Let $\mathcal{F}$ be a space of functions and let

$$p^{in} = \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$$

be the empirical distribution of the data. Then $\mathcal{F}^\star$ is defined to be the space of linear functionals $\mu : \mathcal{F} \to \mathbb{R}$ such that there exists $d \in \mathcal{F}$ satisfying

$$\mu(f) = \langle d, f \rangle_{p^{in}}$$

for all $f \in \mathcal{F}$.

Remark 1.9. Such an element $d \in \mathcal{F}$ is in general not unique, because elements of $\mathcal{F}^\star$ only depend on the data. As a consequence, one can define an equivalence relation on $\mathcal{F}$ with

$$f, g \in \mathcal{F} : f \sim g \iff \langle f, \cdot \rangle_{p^{in}} = \langle g, \cdot \rangle_{p^{in}}.$$

Hence $\mathcal{F}/\sim$ equipped with $\langle \cdot,\cdot \rangle_{p^{in}}$ is a finite-dimensional inner product space, thus a Hilbert space and clearly $\mathcal{F}^\star \cong \mathcal{F}/\sim$.

1.4 Derivative of a functional

We mention the canonical definition of a derivative of a functional when $\mathcal{F} = L^2(\mathbb{R}^n)$ and justify its existence.

Definition 1.10 (Functional derivative)**.** [3] Let $C : \mathcal{F} \to \mathbb{R}$ be a functional. The functional derivative of C at $f \in \mathcal{F}$ is the element $\frac{\delta C}{\delta f} \in \mathcal{F}$ such that

$$\int_{\mathbb{R}^n} \frac{\delta C}{\delta f}(x)^T \phi(x) dx = \lim_{\epsilon \searrow 0} \frac{C(f + \epsilon\phi) - C(f)}{\epsilon}$$

if the limit exists for all $\phi \in L^2(\mathbb{R}^n)$. This is uniquely defined since the expression on the left is just the inner product in $L^2(\mathbb{R}^n)$.

Remark 1.11. In the setting of the article [4], the scalar product $\langle \cdot, \cdot \rangle_{p^{in}}$, as defined in the previous section, is used instead of the L^2 inner product with respect to the Lebesgue measure. Hence, in the context where $\mathcal{F}$ is equipped with $\langle \cdot, \cdot \rangle_{p^{in}}$, $\frac{\delta J}{\delta f}$ is an element of $\mathcal{F}$ such that for all $\phi \in C_c(\mathbb{R}^{n_0})$,

$$\left\langle \frac{\delta C}{\delta f}, \phi \right\rangle_{p^{in}} = \lim_{\epsilon \searrow 0} \frac{C(f + \epsilon\phi) - C(f)}{\epsilon}.$$

Theorem 1.12 (Riesz-Markov-Kakutani)**.** [7] Let X be a locally compact Hausdorff space and let $I : C_c(X) \to \mathbb{R}$ be a positive linear functional on the space of continuous, compactly supported functions on X. Then there exists a unique regular measure μ on $(X, \mathcal{B}(X))$ such that

$$\forall \phi \in C_c(X) : I(\phi) = \int_X \phi d\mu.$$

Proposition 1.13. If the linear functional

$$\phi \mapsto \lim_{\epsilon \searrow 0} \frac{C(f + \epsilon\phi) - C(f)}{\epsilon}$$

exists and is positive then the functional derivative $\frac{\delta C}{\delta f} \in \mathcal{F}$ as in Remark 1.11 exists.

Proof. We first note that

$$\phi \mapsto \lim_{\epsilon \searrow 0} \frac{C(f + \epsilon\phi) - C(f)}{\epsilon} = \frac{d}{d\epsilon} C(f + \epsilon\phi)\bigg|_{\epsilon=0} \tag{1}$$

and writing $\phi = \phi_1 + \phi_2$, an application of the total derivative gives

$$\frac{d}{d\epsilon} C(f + \epsilon\phi_1 + \epsilon\phi_2)\bigg|_{\epsilon=0} = \frac{d}{d\epsilon} C(f + \epsilon\phi_1)\bigg|_{\epsilon=0} + \frac{d}{d\epsilon} C(f + \epsilon\phi_2)\bigg|_{\epsilon=0},$$

proving that (1) is indeed a linear functional on $C_c(X)$. Then by the above theorem, there exists a unique Borel measure μ such that for all $\phi \in C_c$,

$$\int \phi d\mu = \lim_{\epsilon \searrow 0} \frac{C(f + \epsilon\phi) - C(f)}{\epsilon}.$$

Since C is only dependent on the values of the input at the data points, the measure μ has to be supported on the data only. This means $\mu = \sum_{i=1}^N \alpha_i \delta_{x_i}$ for some coefficients α_i where δ_{x_i} are Dirac measures at x_i. Then $\frac{\delta C}{\delta f}(x) = \sum_{i=1}^N N\alpha_i I_{\{x = x_i\}}$ is the unique element of $\mathcal{F}$ satisfying the property in Remark 1.11, since

$$\int \phi d\mu = \sum \alpha_i \phi(x_i) = \frac{1}{N} \sum N\alpha_i \phi(x_i) = \left\langle \frac{\delta C}{\delta f}, \phi \right\rangle_{p^{in}}.$$

□

Remark 1.14 (Proof by Riesz Representation Theorem)**.** Alternatively, the existence of $\frac{\delta C}{\delta f}$ follows from the Riesz Representation Theorem, since $\mathcal{F}/\sim$ equipped with the inner product $\langle \cdot, \cdot \rangle_{p^{in}}$ is a finite dimensional Hilbert space as explained in Remark 1.9. The map

$$\phi \mapsto \lim_{\epsilon \searrow 0} \frac{C(f+\epsilon\phi) - C(f)}{\epsilon}$$

is linear and therefore bounded.

Hence the Riesz Representation Theorem implies that there exists $\frac{\delta C}{\delta f} \in \mathcal{F}$, not necessarily unique, such that

$$\lim_{\epsilon \searrow 0} \frac{C(f+\epsilon\phi) - C(f)}{\epsilon} = \left\langle \phi, \frac{\delta C}{\delta f} \right\rangle_{p^{in}}.$$

Definition 1.15 (Data-dependent functional derivative)**.** [4] For a cost functional C define

$$\begin{aligned} \partial_f^{in} C : \mathcal{F} &\to \mathcal{F}^\star \\ g &\mapsto \left\langle \cdot, \frac{\delta C}{\delta g} \right\rangle_{p^{in}} \end{aligned}$$

to represent the functional derivative as an element of $\mathcal{F}^\star$ as defined in Definition 1.8. As a shorthand notation, denote $d|_g := \frac{\delta C}{\delta g} \in \mathcal{F}$.
If g is time dependent, denote the functional derivative as $d|_{g(t)}$.

Remark 1.16. The emphasis of Definition 1.15 is that, naturally, the derivative of the functional should only be dependent on the data and not a particular $\frac{\delta C}{\delta g} \in \mathcal{F}$ that is chosen to represent $\partial_f^{in} C(g)$, since $\frac{\delta C}{\delta g}$ is a function on the entire domain $\mathbb{R}^{n_0}$.

2 Neural tangent kernel asymptotics

2.1 Kernel gradient descent

Definition 2.1 (Kernel gradient)**.** [4] For any multidimensional kernel K define the mapping

$$\begin{aligned} \Phi_K : \mathcal{F}^\star &\to \mathcal{F} \\ \mu &\mapsto f_\mu \end{aligned}$$

where, letting $\mu = \langle d, \cdot \rangle_{p^{in}}$ for some $d \in \mathcal{F}$,

$$f_{\mu,i}(x) := \langle d, K_{i,\cdot}(x, \cdot) \rangle_{p^{in}}$$

is defined to be the i-th component of $f_\mu(x)$. Then the kernel gradient of a functional C at $h \in \mathcal{F}$ with respect to a kernel K is defined as

$$\nabla_K C(h) := \Phi_K(\partial_f^{in} C(h)).$$

A function $g : [0, \infty) \to \mathcal{F}$ is said to follow the kernel gradient descent with respect to a kernel K and a cost functional C if it satisfies

$$\partial_t g(t) = -\nabla_K C(g(t)) = -\Phi_K(\partial_f^{in} C(g(t))).$$

Remark 2.2. The preceding equation can be written more explicitly as

$$\partial_t g(t) = -\Phi_K(\langle d|_{g(t)}, \cdot \rangle_{p^{in}}) = -\frac{1}{N} \sum_{i=1}^{N} K(x, x_i) d|_h(x_i),$$

where the second equality follows directly from the definition of Φ_K.
The mapping Φ_K is therefore a way of extending a function that is only defined on the data to a function on the entire domain $\mathbb{R}^{n_0}$.

Remark 2.3. In our setting of finite width, dense network functions, we have $g(t) = F^{(L)}(\theta(t))$, also denoted as $f_{\theta(t)}$. Then, according to these definitions, $f_{\theta(t)} := F^{(L)}(\theta(t))$ follows the kernel gradient descent with respect to K if

$$\partial_t f_{\theta(t)} = -\nabla_K C(f_{\theta(t)}).$$

It remains to specify a suitable kernel K for the kernel gradient descent. As shown in Proposition 2.5, there exists a natural choice for this kernel that corresponds to the classical gradient descent on the parameter space.

Definition 2.4 (Neural tangent kernel)**.** Assume the usual dense network setup and recall that

$$F^{(L)} : \mathbb{R}^P \to \mathcal{F}$$

maps a parameter vector to its corresponding network function. The multidimensional kernel

$$\begin{aligned} \Theta^{(L)}(\theta) &: \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}^{n_L \times n_L}, \quad \theta \in \mathbb{R}^P \\ \Theta^{(L)}(\theta) &= \sum_{p=1}^{P} \partial_{\theta_p} F^{(L)}(\theta) \otimes \partial_{\theta_p} F^{(L)}(\theta), \end{aligned}$$

is called the neural tangent kernel. The tensor product notation means explicitly

$$\left(\Theta^{(L)}(\theta)(x,y)\right)_{ij} = \sum_{p=1}^{P} \left[\partial_{\theta_p} F^{(L)}(\theta)(x)\right]_i \left[\partial_{\theta_p} F^{(L)}(\theta)(y)\right]_j .$$

The next proposition shows that the gradient descent on $C \circ F^{(L)}$ corresponds to the kernel gradient descent with respect to the parametrized family of kernels defined above.

Proposition 2.5 (Kernel gradient descent representation)**.** [4] Let $f_{\theta(t)} := F^{(L)}(\theta(t))$ and suppose this function is trained using the gradient descent on the network parameters,

$$\partial_t \theta = -\nabla_\theta C(f_\theta)$$

that is, componentwise,

$$\partial_t \theta_p = -\langle \partial_f^{in} C(f_\theta), \partial_{\theta_p} f_\theta \rangle_{p^{in}} = -\langle d|_{f_\theta}, \partial_{\theta_p} f_\theta \rangle_{p^{in}}.$$

Then

$$\partial_t f_{\theta(t)}(x) = -\nabla_{\Theta^{(L)}} C(f_\theta).$$

Proof. By the chain rule,

$$\begin{aligned} \partial_t f_{\theta(t)}(x) &= \sum_{p=1}^{P} \partial_{\theta_p} f_{\theta(t)}(x) \partial_t \theta_p(t) \\ &= -\sum_{p=1}^{P} \langle d|_{f_{\theta(t)}}, \partial_{\theta_p} f_{\theta(t)} \rangle_{p^{in}} \partial_{\theta_p} f_{\theta(t)}(x) \\ &= -\frac{1}{N} \sum_{p=1}^{P} \sum_{i=1}^{N} \partial_{\theta_p} f_\theta(x) \partial_{\theta_p} f_\theta(x_i)^T d|_{f_\theta}(x_i) \\ &= -\Phi_{\Theta^{(L)}}(\partial_f^{in} C(f_\theta)) = -\nabla_{\Theta^{(L)}} C(f_\theta) \end{aligned}$$

with $\Theta^{(L)} = \partial_{\theta_p} f_\theta \otimes \partial_{\theta_p} f_\theta$ as required. □

Given this correspondence between the vanilla gradient descent and the kernel gradient descent, naturally one might ask under what circumstances the kernel gradient descent converges to a global minimum. As the following result shows, positive-definiteness of the kernel is a sufficient condition.

Lemma 2.6 (A chain rule)**.** Let $C : \mathcal{F} \to \mathbb{R}$ be a functional, $\theta : [0, \infty) \to \mathbb{R}^P$ and $F : \mathbb{R}^P \to \mathcal{F}$. Then

$$\partial_t (C \circ F)(\theta(t)) = \langle \partial_f^{in} C(F(\theta(t))), \partial_t F(\theta(t)) \rangle_{p^{in}}.$$

Proposition 2.7 (Evolution of the cost functional)**.** [4] If $f_{\theta(t)} := F^{(L)}(\theta(t))$ evolves according to the kernel gradient descent with respect to the neural tangent kernel, i.e. if

$$\partial_t f_{\theta(t)} = -\nabla_{\Theta^{(L)}} C(f_{\theta(t)})$$

then

$$\partial_t C(f_{\theta(t)}) = -\langle d|_{f_\theta}, \nabla_{\Theta^{(L)}} C(f_{\theta(t)}) \rangle_{p^{in}}.$$

Proof. An application of the above chain rule yields

$$\partial_t C(f_{\theta(t)}) = \langle \partial_f^{in} C|_{f_\theta}, \partial_t f_{\theta(t)} \rangle_{p^{in}} = -\langle \partial_f^{in} C|_{f_\theta}, \nabla_{\Theta^{(L)}} C(f_\theta) \rangle_{p^{in}} = -\langle d|_{f_{\theta(t)}}, \nabla_{\Theta^{(L)}} C(f_{\theta(t)}) \rangle_{p^{in}}.$$

□

Corollary 2.8 (Cost functional decay)**.** [4] If the kernel $\Theta^{(L)}$ is positive definite with respect to $\|\cdot\|_{p^{in}}$ then for all $t \geq 0$

$$\partial_t C(f_\theta) \leq 0.$$

Proof. By the proposition and definition of the kernel gradient,

$$\partial_t C(f_{\theta(t)}) = -\langle d|_{f_\theta}, \nabla_{\Theta^{(L)}} C(f_{\theta(t)}) \rangle_{p^{in}} = -\sum_{i=1}^{N} \sum_{j=1}^{N} d|_{f_\theta}(x_i)^T \Theta^{(L)}(x_i, x_j) d|_{f_\theta}(x_j) = -\|d|_{f_\theta}\|^2_{\Theta^{(L)}}.$$

This is non-negative, since $\|d|_{f_\theta}\|^2_{p^{in}} \geq 0$ and $\Theta^{(L)}$ is positive definite with respect to $\|\cdot\|_{p^{in}}$.

□

Remark 2.9. This gives us hope for the convergence of the kernel gradient descent to a global minimum of the cost functional, because in fact $-\|d|_{f_\theta}\|^2_{\Theta^{(L)}} < 0$ unless

$$\|d|_{f_\theta}\|_{p^{in}} = 0$$

i.e. $d|_{f_\theta}(x_i) = 0$ for all $i = 1, \ldots, N$. While the kernel $\Theta^{(L)}$ is random because of the random initialization, it will turn out that the limiting kernel as $n_1, \ldots, n_L \to \infty$ is deterministic.

2.2 Infinite width limit of the neural tangent kernel

The following is the main result of the work [4], showing the convergence of the random kernels $\Theta^{(L)}$ to a deterministic kernel as the width of each hidden layer increases to infinity sequentially over the total number of hidden layers. It uses the asymptotic result in Proposition 1.3, with the proof in the appendix. In the following, let Id_{n_L} denote a $n_L \times n_L$ identity matrix and let $\dot{\sigma}$ denote the derivative of σ.

Theorem 2.10 (Convergence in probability)**.** [4] Let $n_1, ..., n_{L-1} \to \infty$ sequentially. Then for any $x, x' \in \mathbb{R}^{n_0}$,

$$\Theta^{(L)}(x, x') \xrightarrow{\mathrm{P}} \Theta^{(L)}_\infty(x, x') \otimes \mathrm{Id}_{n_L},$$

where $\Theta_\infty^{(L)} : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$ is defined recursively by

$$\begin{aligned}\Theta_\infty^{(1)}(x,x') &= \Sigma^{(1)}(x,x')\\ \Theta_\infty^{(\ell+1)}(x,x') &= \Theta_\infty^{(\ell)}(x,x')\dot{\Sigma}^{(\ell+1)}(x,x') + \Sigma^{(\ell+1)}(x,x'),\end{aligned}$$

and where

$$\dot{\Sigma}^{(\ell+1)}(x,x') = \mathbb{E}_{(u,v)\sim\mathcal{N}\left(0,\Lambda^{(\ell)}(x,x')\right)}\left[\dot{\sigma}(v)\dot{\sigma}(v)\right] + \beta^2,$$

with $\Lambda^{(\ell)}(x,x')$ defined as in Proposition 1.3.

Remark 2.11. Note that the kernels are only dependent on the layer and node numbers and the activation function, while they are independent of training time (since there is no training taking place), thus independent of the cost functional. Moreover, the limiting kernel $\Theta_\infty^{(L)} \otimes \mathrm{Id}_{n_L}$ is deterministic.

Theorem 2.12 (Uniform convergence in training)**.** [4] Assume that the activation σ is Lipschitz and almost everywhere twice differentiable with bounded second derivative and let $T > 0$ be such that $\int_0^T d|_{f_{\theta(t)}} dt$ is stochastically bounded. If the kernel $\Theta^{(L)}$ evolves in time along with $\theta(t)$ having the dynamics of the gradient descent, i.e.

$$\Theta^{(L)}(t) := \sum_{p=1}^{P} \partial_{\theta_p} F^{(L)}(\theta(t)) \otimes \partial_{\theta_p} F^{(L)}(\theta(t))$$

with $\theta(t)$ satisfying

$$\partial_t \theta(t) = -\nabla_\theta C(\theta(t)),$$

then

$$\Theta^{(L)}(t) \to \Theta_\infty^{(L)} \otimes \mathrm{Id}_{n_L}$$

uniformly on $t \in [0,T]$ as $n_1, \ldots, n_{L-1} \to \infty$.

Remark 2.13. This result builds on Theorem 2.10 and, by contrast, depends on the training dynamics, therefore on the cost functional.

Corollary 2.14. [4] In the infinite width limit, the network function f_t satisfies the dynamics

$$\partial_t f_t = \Phi_{\Theta_\infty^{(L)} \otimes \mathrm{Id}_{n_L}}(\langle -d|_{f_t}, \cdot \rangle_{p^{in}}).$$

Proof. This is a direct consequence of Proposition 2.5 and Theorem 2.12. □

Remark 2.15. In the infinite width limit, it no longer makes practical sense to speak of training f on the parameter space, since this space is now infinite-dimensional. Hence the notation f_t instead of $f_{\theta(t)}$ which was used in the case of finite width layers.

Proposition 2.16 (Positive-definiteness of the NTK)**.** [4] If the activation σ is Lipschitz and non-polynomial, the restricted NTK

$$\Theta_\infty^{(L)} : \mathbb{S}^{n_0-1} \times \mathbb{S}^{n_0-1} \to \mathbb{R}^{n_L \times n_L}$$

is positive-definite with respect to $\|\cdot\|_{p^{in}}$, where $\mathbb{S}^{n_0-1} := \{x \in \mathbb{R}^{n_0} : \|x\|_2 = 1\}$.

Remark 2.17. Due to Corollary 2.8, the proposition thus gives a condition under which the kernel gradient descent with respect to the neural tangent kernel converges to a global minimum in the infinite width limit.

3 Explicit solution for least squares

Having presented the results of [4] in full generality in Section 2, we now focus on a special case where $n_L = 1$, σ is the ReLu activation function $\sigma(x) := \max\{0, x\}$ and C is the least squares cost functional defined below.

Definition 3.1 (Least squares cost functional)**.** Consider a goal function $f^* \in \mathcal{F}$ and for any $f \in \mathcal{F}$ set

$$C(f) = \frac{1}{2}||f - f^*||^2_{p^{in}} = \frac{1}{2}\mathbb{E}_{x \sim p^{in}}\left[\|f(x) - f^*(x)\|^2\right]. \tag{2}$$

Remark 3.2. While we state Definition 3.1 in terms of a goal function $f^* \in \mathcal{F}$, $C(f)$ only depends on the values f^* takes at $x_1, ..., x_N$. If we are provided with training outputs $\{y_i\}_{i=1}^N \subset \mathbb{R}^{N_L}$, then it is natural for $f^* \in \mathcal{F}$ to be a function satisfying $f^*(x_i) = y_i$ for all $i = 1, ..., N$.

The reason why we are considering the setup with a ReLu activation function and a least squares cost functional is that it allows us to

- derive an analytic expression for the pointwise $t \to \infty$ limit of f solving the differential equation given by Corollary 2.14,
- derive an analytic recursion for the limiting neural tangent kernel given by Theorem 2.10.

Throughout this section $K := \Theta^{(L)}_\infty \otimes \mathrm{Id}_{n_L} = \Theta^{(L)}_\infty$ for economy of notation.

3.1 Functional derivative of the cost functional

To apply Corollary 2.14 in our setup, we must first derive $\frac{\delta C}{\delta f}$.

Proposition 3.3. [4] The functional $C(f) = \frac{1}{2}\|f - f^*\|^2_{p^{in}}$ has a functional derivative

$$\frac{\delta C}{\delta f} = f - f^*$$

for any $f \in \mathcal{F}$.

Proof. Following the definition of the functional derivative, we have

$$C(f + \epsilon\phi) = \frac{1}{2}\mathbb{E}_{x \sim p^{in}}\left[\|f(x) - f^*(x)\|^2 + 2\epsilon(f(x) - f^*(x))^T\phi(x) + \epsilon^2\|\phi(x)\|^2\right]$$

and so

$$\frac{C(f + \epsilon\phi) - C(f)}{\epsilon} = \mathbb{E}_{x \sim p^{in}}\left[(f(x) - f^*(x))^T\phi(x)\right] + \frac{\epsilon}{2}\mathbb{E}_{x \sim p^{in}}\left[\|\phi(x)\|^2\right].$$

Clearly

$$\lim_{\epsilon \searrow 0}\frac{C(f + \epsilon\phi) - C(f)}{\epsilon} = \mathbb{E}_{x \sim p^{in}}\left[(f(x) - f^*(x))^T\phi(x)\right] = \langle f - f^*, \phi\rangle_{p^{in}}$$

for all $\phi \in \mathcal{F}$. □

Remark 3.4. Note that since $\| \cdot \|_{p^{in}}$ takes into account only the data points rather than the entire domain $\mathbb{R}^{n_0}$, $\frac{\delta C}{\delta f}$ is not uniquely defined as an element of $\mathcal{F}$.

3.2 Kernel gradient descent

Before attempting to find the pointwise $t \to \infty$ limit of f solving the kernel gradient descent

$$\partial_t f_t = \Phi_K \left(\langle -d|_{f_t}, \cdot \rangle_{p^{in}} \right).$$

we must introduce some notation. This notation will allow us to derive a useful representation of f_t at any $t \geq 0$.

Definition 3.5. Let $\Pi : \mathcal{F} \to \mathcal{F}$ be given by

$$\Pi(f) = \Phi_K \left(\langle f, \cdot \rangle_{p^{in}} \right)$$

and let $e^{-t\Pi} : \mathcal{F} \to \mathcal{F}$ be given by

$$e^{-t\Pi}(f) = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!} \Pi^k(f)$$

for any $t \geq 0$.

Lemma 3.6. Let the cost functional $C(f)$ be given by (2). Then the solution of the differential equation

$$\partial_t f_t = \Phi_K \left(\langle f^* - f_t, \cdot \rangle_{p^{in}} \right)$$

with initial condition $f_0 : \mathbb{R}^{n_0} \to \mathbb{R}$ can be written as

$$f_t(x) = f^*(x) + e^{-t\Pi}(f_0 - f^*)(x). \tag{3}$$

Proof. Taking the partial derivative of (3) with respect to t, we have

$$\partial_t f_t(x) = -\sum_{k=0}^{\infty} \frac{(-t)^k}{k!} \Pi^{k+1}(f_0 - f^*)(x).$$

We also have

$$\Pi(f^* - f)(x) = \Pi \left(-\sum_{k=0}^{\infty} \frac{(-t)^k}{k!} \Pi^k(f_0 - f^*) \right)(x) = -\sum_{k=0}^{\infty} \frac{(-t)^k}{k!} \Pi^{k+1}(f_0 - f^*)(x)$$

by linearity of Π. Thus (3) satisfies

$$\partial_t f_t(x) = \Pi(f^* - f)(x)$$

and setting $t = 0$ we confirm that f_t has the required initial condition. □

3.3 Limiting solution

Finally we are in a position to derive the limiting solution which corresponds to the completion of training. Once again, this requires some notation.

Definition 3.7. We define an $N \times N$-dimensional Gram matrix $\tilde{K}$ with entries

$$\tilde{K}_{ij} = K(x_i, x_j),$$

where $i, j \in \{1, ..., N\}$.

Definition 3.8. [4] We define a map $\kappa : \mathbb{R}^{n_0} \to \mathbb{R}^N$ and N-dimensional vectors y^* and y_0 through

$$\kappa_i(x) = K(x, x_i), \qquad y_i^* = f^*(x_i), \qquad y_{0,i} = f_0(x_i),$$

where $i \in \{1, ..., N\}$.

Proposition 3.9. [4] Provided the Gram matrix $\tilde{K}$ is invertible,

$$f_\infty(x) := \lim_{t\to\infty} f_t(x) = \kappa^T(x)\tilde{K}^{-1}y^* + \left(f_0(x) - \kappa^T\tilde{K}^{-1}y_0\right), \tag{4}$$

for any $x \in \mathbb{R}^{n_0}$.

Proof. First we note the representation

$$\Pi(f)(x) = \frac{1}{N}\sum_{i=1}^{N} f(x_i)K(x_i, x).$$

We deduce from it that the range of Π is equivalent to the span of the N functions $g_i : \mathbb{R}^{n_0} \to \mathbb{R}$ given by

$$g_i(x) = K(x_i, x)$$

indexed by $i = 1, ..., N$. [9] In particular, Π has $N^* \leq N$ positive eigenvalues $\lambda_1, ..., \lambda_{N^*}$ and no negative eigenvalues. Since $f^* - f_0 \in \mathcal{F}$ we can write

$$f^* - f_0 = \Delta^0 + \Delta^1 + ... + \Delta^{N_*},$$

where Δ^0 is in the kernel of Π and where Δ^j is in the j^{th} eigenspace of Π for $j = 1, ..., N^*$. This allows us to write

$$f_t(x) = f^*(x) + \Delta^0(x) + \sum_{j=1}^{N^*} \exp(-t\lambda_j)\Delta^j(x)$$

from which it is clear that

$$\lim_{t\to\infty} f_t(x) = f^*(x) + \Delta^0(x) = f_0(x) - \sum_{j=1}^{N^*}\Delta^j(x).$$

Since the span of the eigenfunctions of Π corresponding to positive eigenvalues is identical to the span of $g_1, ..., g_N$, necessarily

$$\sum_{j=1}^{N^*}\Delta^j(x) = \kappa^T(x)\alpha^*$$

for some $\alpha^* \in \mathbb{R}^N$. Since we are interested in finding an orthogonal projection of $f^* - f_0$ onto the span of $g_1, ..., g_N$, this α^* is obtained by minimizing

$$\frac{1}{2}||f^* - f_0 - \kappa^T\alpha||^2_{p^{in}}$$

over $\alpha \in \mathbb{R}^N$. This is a linear regression problem which yields

$$\alpha^* = \tilde{K}^{-1}(y^* - y_0)$$

provided $\tilde{K}$ is invertible. □

Remark 3.10. When the initialization function f_0 is identically zero, (4) simplifies to

$$f_\infty(x) = \kappa^T(x)\tilde{K}^{-1}y^* \tag{5}$$

and this is the case we will be considering while stating convergence results and performing numerics.

3.4 Co-variance recursion for ReLu activation

The main reason for considering a ReLu activation function is that it allows us to derive an analytic recursion for the neural tangent kernel. Essentially, the results stated in Theorem 3.12 follow from the Cholesky decomposition

$$\Lambda^{(\ell)}(x,x') = L^{(\ell)}(x,x')(L^{(\ell)}(x,x'))^T$$

with

$$L^{(\ell)}(x,x') = \begin{pmatrix} \sqrt{\Sigma^{(\ell)}(x,x)} & 0 \\ \frac{\Sigma^{(\ell)}(x',x)}{\sqrt{\Sigma^{(\ell)}(x,x)}} & \sqrt{\Sigma^{(\ell)}(x',x') - \frac{(\Sigma^{(\ell)}(x',x))^2}{\Sigma^{(\ell)}(x,x)}} \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

and from analytic expressions for first and second order arc-cosine kernel function provided by [2]. Nevertheless, before presenting the theorem we must introduce some notation.

Definition 3.11. For any $\ell \in \{0, ..., L-1\}$ we introduce the maps $\rho^{(\ell)}, r^{(\ell)} : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$ given by

$$\rho^{(\ell)}(x,x') = \frac{\Lambda^{(\ell)}_{12}(x,x')}{\sqrt{\Lambda^{(\ell)}_{11}(x,x')\Lambda^{(\ell)}_{22}(x,x')}}, \qquad r^{(\ell)}(x,x') = \sqrt{\Lambda^{(\ell)}_{11}(x,x')\Lambda^{(\ell)}_{22}(x,x')},$$

where $\Lambda^{(\ell)}$ is defined as in Proposition 1.3.

Theorem 3.12. Let σ be the ReLu activation function. Then for any $\ell = 0, ..., L-1$

$$\Sigma^{(\ell+1)}(x,x') = \frac{1}{2\pi} r^{(\ell)}(x,x')\left(\sin\left(\cos^{-1}\left(\rho^{(\ell)}(x,x')\right)\right) + \left(\pi - \cos^{-1}\left(\rho^{(\ell)}(x,x')\right)\right)\rho^{(\ell)}(x,x')\right) + \beta^2 \quad (6)$$

and

$$\dot{\Sigma}^{(\ell+1)}(x,x') = \frac{1}{2\pi}\left(\pi - \cos^{-1}\left(\rho^{(\ell)}(x,x')\right)\right). \quad (7)$$

Since $\Sigma^{(\ell+1)}(x,x')$ and $\dot{\Sigma}^{(\ell+1)}(x,x')$ have a non-trivial dependence on $\rho^{(\ell)}(x,x')$ and $r^{(\ell)}(x,x')$, we do not attempt to state a closed-form formula for $\Theta^{(\ell+1)}_\infty(x,x')$.

3.5 Discussion of the NTK recursion with ReLu activation

We would now like to make a few remarks about the forms of $\Sigma^{(\ell+1)}(x,x')$ and $\dot{\Sigma}^{(\ell+1)}(x,x')$ presented in Section 3.4 since their properties extend to properties of the limiting neural tangent kernel. Given the fact that $\rho^{(\ell)}(x,x') \in [-1,1]$, it is straightforward to see that (6) and (7) satisfy

$$\beta^2 \leq \Sigma^{(\ell+1)}(x,x') \leq \frac{1}{2} r^{(\ell)}(x,x') + \beta^2, \qquad 0 \leq \dot{\Sigma}^{(\ell+1)}(x,x') \leq \frac{1}{2}.$$

We also observe that

$$\Sigma^{(\ell+1)}(x,x) = \frac{1}{2}\Sigma^{(\ell)}(x,x) + \beta^2 = \frac{1}{2^\ell}\Sigma^{(1)}(x,x) + \beta^2 \sum_{i=1}^{\ell} \frac{1}{2^{i-1}} \leq \Sigma^{(1)}(x,x) + 2\beta^2.$$

Now,

$$\beta^2 \leq \Theta^{(\ell+1)}_\infty(x,x') = \Theta^{(\ell)}_\infty(x,x')\dot{\Sigma}^{(\ell+1)}(x,x') + \Sigma^{(\ell+1)}(x,x') \leq \frac{1}{2}\Theta^{(\ell)}_\infty(x,x') + \frac{1}{2} r^{(\ell)}(x,x') + \beta^2$$

and we deduce that

$$\begin{aligned}\beta^2 \leq \Theta_\infty^{(\ell+1)}(x,x') &\leq \frac{1}{2^\ell}\Theta_\infty^{(1)}(x,x') + \beta^2 \sum_{i=1}^{\ell} \frac{1}{2^{i-1}} + \sum_{i=1}^{\ell} \frac{1}{2^{\ell-i+1}} r^{(i)}(x,x') \\ &\leq \Theta_\infty^{(1)}(x,x') + 2\beta^2 + \sqrt{\left(\Sigma^{(1)}(x,x)+2\beta^2\right)\left(\Sigma^{(1)}(x',x')+2\beta^2\right)}.\end{aligned} \tag{8}$$

The significance of this result, is that we obtained uniform (in ℓ) bounds for the limiting neural tangent kernel at any depth. An interesting case occurs when we set $\beta = 0$. Then,

$$r^{(\ell+1)}(x,x') = \frac{1}{2^\ell}\sqrt{\Sigma^{(1)}(x,x)\Sigma^{(1)}(x',x')}$$

and we have

$$0 \leq \Theta_\infty^{(\ell+1)}(x,x') \leq \frac{1}{2^\ell}\Theta_\infty^{(1)}(x,x') + \frac{\ell}{2^\ell}\sqrt{\Sigma^{(1)}(x,x)\Sigma^{(1)}(x',x')}$$

In particular, as $\ell \to \infty$, we observe an exponential decay of $\Theta_\infty^{(\ell+1)}(x,x')$. Finally, let us consider the case when $\beta = 0$ and $n_0 = 1$ which as we will see leads to a piecewise linear limiting neural tangent kernel. Indeed, if $x, x' \in \mathbb{R} \setminus \{0\}$ have the same sign then $\rho^{(1)}(x,x') = 1$ and otherwise $\rho^{(1)}(x,x') = -1$. In the case of $\rho^{(1)}(x,x') = 1$, this leads to

$$\Theta_\infty^{(\ell+1)}(x,x') = \frac{1}{2}\Theta_\infty^{(\ell)}(x,x') + \frac{1}{2^\ell}x\cdot x' = \frac{1}{2^\ell}\Theta_\infty^{(1)}(x,x') + \frac{\ell}{2^\ell}x\cdot x' = \frac{\ell+1}{2^\ell}x\cdot x'.$$

In the case of $\rho^{(1)}(x,x') = -1$, both $\Sigma^{(\ell)}(x,x')$ and $\dot{\Sigma}(x,x')$ vanish and we have

$$\Theta_\infty^{(\ell+1)}(x,x') = 0.$$

Thus, for fixed x (or x'), $\Theta_\infty^{(\ell+1)}(x,x')$ is piecewise linear with a kink at $x' = 0$ (or $x = 0$). The significance of this result is that for $\beta = 0$ and $n_0 = 1$, the gram matrix $\tilde{K}$ is non-invertible for any $N \geq 3$. This observation together with the stabilizing effect of β on the limiting neural tangent kernel deduced from (8) motivate the use of $\beta > 0$.

4 Convergence rate results

While the work [4] establishes convergence in probability of the neural tangent kernel to a deterministic limiting kernel and also the limiting dynamics of a network function trained using gradient descent, the following problems are not examined:

- the approximation of a neural tangent kernel by the deterministic limiting kernel for a network of sufficiently large width,
- the approximation of a network function trained using kernel gradient descent with respect to the limiting kernel by the function trained using gradient descent for a network of sufficiently large width.

These notions are addressed in [1] for a particular case of the model presented in Section 1.1 with $\beta = 0$, $n_L = 1$ and a ReLu activation function.

Remark 4.1. Apart from the restrictions that $\beta = 0$ and $n_L = 1$, the results of [1] are stated for a neural network with $a = 2$ in the activations

$$\sqrt{\frac{a}{n_\ell}}W^{(\ell)}\alpha^{(\ell)}(x;\theta), \quad l = 0,\ldots,L-1,$$

whereas our setup assumes that $a = 1$. This inconsistency, however, is easily resolved given that for a Relu activation function σ, we have $\sigma(bx) = b\sigma(x)$ for any $b \geq 0$. Thus, the network function of [1] is obtained from our network function by scaling by $2^{(1-L)/2}$. Another inconsistency between our setup and that of [1] is that we index layers by $\ell = 0, ..., L$ whereas [1] index layers by $\ell = 1, ..., L$. With these two inconsistencies resolved, the results of [1] in our setup are stated in Theorem 4.2 and Theorem 4.4 below.

4.1 Neural tangent kernel convergence

Theorem 4.2. [1] For any $\epsilon > 0$ and $\delta \in (0,1)$, let

$$\min_{l\in\{1,\ldots,L-1\}} n_l \geq \Omega\left(\frac{(L-1)^6}{\epsilon^4}\log\left(\frac{(L-1)}{\delta}\right)\right).$$

Then for any inputs $x, x' \in \mathbb{R}^{n_0}$ such that $||x|| \leq 1, ||x'|| \leq 1$, with probability at least $1-\delta$ we have:

$$\left|\Theta^{(L)}(x,x';\theta) - \Theta_\infty^{(L)}(x,x';\theta)\right| \leq \frac{L}{n_0}2^{1-L}\epsilon.$$

Remark 4.3. The Ω notation stands for minimal asymptotic growth. This assumption means that the sequences $n_1,\ldots,n_{L-1}$ are such that $\exists N \in \mathbb{N}, \xi > 0, C > 0$ such that $\forall L \geq N, \forall \epsilon, \delta < \xi$:

$$\min_{l\in\{1,\ldots,L-1\}} n_l \geq C\left(\frac{(L-1)^6}{\epsilon^4}\log\left(\frac{(L-1)}{\delta}\right)\right)$$

Intuitively speaking, one expects the required number of nodes in the hidden layers to increase as we introduce more layers (L increases), or if we require closer convergence (ϵ decreases closer to 0) with higher probability (δ decreases closer to 0).

4.2 Gradient descent convergence

In the case of least-squares regression, we assume that f_0 is identically zero. For a neural network with finite widths $n_1, ..., n_{L-1}$, we let $f_{\theta(t)} = F^{(L)}(\theta(t))$ be the network function trained using regular gradient descent. That is, it holds that

$$\partial_t\theta(t) = -\nabla_\theta C(f_\theta).$$

For any $x \in \mathbb{R}^{n_0}$, we let $f_{\lim}(x)$ denote $\lim_{t\to\infty} f_{\theta(t)}(x)$. As before, we let f_∞ denote the pointwise $t \to \infty$ limit of a function f trained using kernel gradient descent with kernel $\Theta_\infty^{(L)}$. In this setup, we have the following convergence result.

Theorem 4.4. [1] For any $\kappa > 0$, suppose

$$1/\kappa = \text{poly}(1/\epsilon, \log(N/\delta))$$

and let $n_1, ..., n_{L-1} = m$ with

$$m \geq \text{poly}(1/\kappa, L-1, 1/\lambda_0, N, \log(1/\delta)).$$

Then for any $x \in \mathbb{R}^{n_0}$ such that $||x|| = 1$, with probability at least $1-\delta$, we have:

$$|\kappa f_{\lim}(x) - f_\infty(x)| \leq \epsilon.$$

5 Numerical experiments

5.1 Least squares kernel regression

In this subsection we present several plots obtained by implementing f_∞ given by (5) when K is the limiting neural tangent kernel corresponding to $\beta = 0.1$, $L = 5$ and $n_L = 1$ for various (simulated) test and training data sets.

Experiment #1: Here we generate training data $\{(x_i, y_i)\}_{i=1}^{200} \in \mathbb{R}\times\mathbb{R}$ by independently sampling $x_i, z_i \sim \mathcal{N}(0,1)$ and setting

$$y_i = x_i^2 + az_i,$$

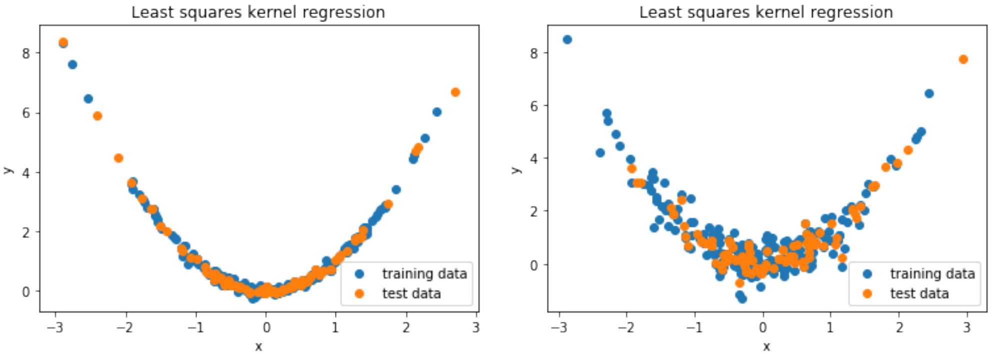

Figure 1: Least squares kernel regression for $a = 0.1$ (left) and $a = 0.5$ (right).

where $a \in \mathbb{R}$ is a constant that determines the noise level of the test data. We then produce test data $\{x_i'\}_{i=1}^{200} \in \mathbb{R}$ by independently samplingn $x_i' \sim \mathcal{N}(0,1)$.

Experiment #2: The setup in this experiment is the same as before except we set

$$y_i = H(x_i) + bz_i,$$

where H is the Heaviside step function and where $b \in \mathbb{R}$ is a constant that determines the noise level of the test data.

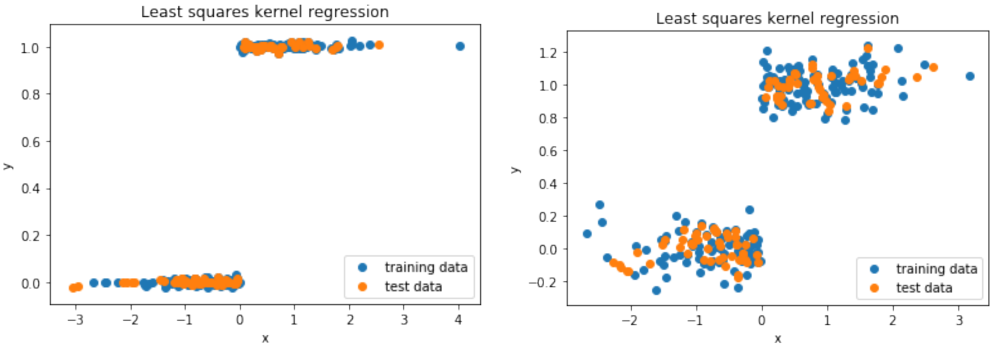

Figure 2: Least squares kernel regression for $b = 0.01$ (left) and $b = 0.1$ (right).

Experiment #3: We now produce test data $\{(x_i, y_i)\}_{i=1}^{200} \in \mathbb{R}^2 \times \mathbb{R}$ by independently sampling $x_{i1}, x_{i2}, z_i \sim \mathcal{N}(0,1)$ and setting

$$y_i = x_{i1}^2 + x_{i2}^2 + cz_i,$$

where $c \in \mathbb{R}$ is a constant that determines the noise level of the test data. We then produce test data $\{x_i'\}_{i=1}^{200} \in \mathbb{R}^2$ by independently simulating $x_{i1}', x_{i2}' \sim \mathcal{N}(0,1)$.

A key observation from Figures 1, 2 and 3 is that our implementation of least squares kernel regression performs much better in the case of less noisy training data. Test data points are assigned values based more on the values of training data points with similar inputs and less on the overall trend of the data. This is consistent with other kernel methods given that a test data point is mapped to a linear combination of training outputs based on the kernel values attained at test and training input pairs. While we must appreciate that kernel methods are non-parametric, we would like to see test data mapped closer to the quadratic or Heaviside trends in Figure 1 for $a = 0.5$ or in Figure 2 for $b = 0.1$.

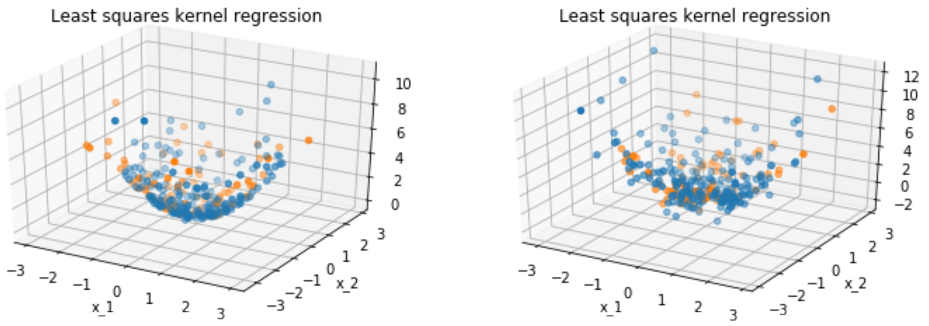

Figure 3: Least squares kernel regression for $c = 0.1$ (left) and $c = 1$ (right).

5.2 Convergence of the empirical NTK

We also perform some numerics to illustrate the result of Theorem 2.10. We start by fixing $n_0 = 4, n_L = 1, \beta = 0$ and $L = 3$. We also select two vectors $x_1, x_2 \in \mathbb{R}^4$ given by

$$x_1 = (0, 1, 3, 8)^T, \qquad\qquad x_2 = (-1, 3, 4, 5)^T$$

and determine the limiting neural tangent kernel $\Theta_\infty^{(L)}(x_1, x_2)$. We would like to compare this value with a histogram of empirical neural tangent kernels $\Theta^{(L)}(\theta)(x_1, x_2)$ corresponding to different widths of the hidden layers. As a reminder, $\Theta^{(L)}(\theta)(x_1, x_2)$ is stochastic because the neural network parameters θ are initiated as i.i.d. samples from a $N(0, 1)$ r.v. To produce the plots below, we set $n_1 = n_2 = d \cdot i$ for $i = 1, ..., 4$ and for some $d \in \mathbb{Z}_+$. In each case, we produce histograms using N_s random initializations.

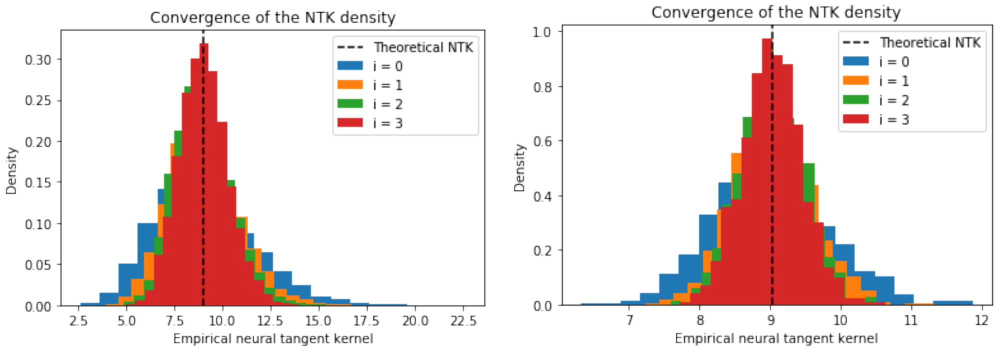

Figure 4: Histograms of empirical neural tangent kernels for $(d, N_s) = (100, 10000)$ (left) and $(d, N_s) = (1000, 1000)$ (right).

Figure 3 is consistent with Theorem 2.10 since we observe that histograms of empirical neural tangent kernels converge to a point mass at approximately 9.025 which is the limiting NTK given by the theorem. We also observe that this convergence is more rapid while increasing i in the case of $d = 100$ than in the case of $d = 1000$ which is consistent with the discussion of numerics in [4].

A Appendix

In the appendix, we provide proofs of Theorem 1.3 and Theorem 2.10 and an auxiliary discussion that, if included in the main document, would distract from more important results and observations. While the proofs are based on [4], we provide more details here.

A.1 Proof of Theorem 1.3

Proof of Theorem 1.3. [4] We will prove the result by induction starting from $L = 1$. In this case,

$$f_\theta(x) = \frac{1}{\sqrt{n_0}} W^{(0)} x + \beta b^{(0)}.$$

For any collection of input vectors $x^1, ..., x^N \in \mathbb{R}^{n_0}$, the n_1 random vectors

$$\left(f_{\theta,k}(x^1), ..., f_{\theta,k}(x^N)\right)^T \tag{9}$$

indexed by $k = 1, ..., n_1$ are i.i.d. since entries in $W^{(0)}$ and in $b^{(0)}$ are independent samples from a $\mathcal{N}(0,1)$ r.v. Moreover (9) is a Gaussian random vector (since each component is a linear combination of draws from a $\mathcal{N}(0,1)$ r.v.) with zero mean vector. Letting $W_1^{(0)}$ and $b^{(0)}$ denote the first row of $W^{(0)}$ and the first element of $b^{(0)}$ respectively, we find that (9) has co-variance $\Sigma^{(1)} \in \mathbb{R}^{N \times N}$ with terms of the form

$$\Sigma_{ij}^{(1)} = \mathbb{E}\left[\left(\frac{1}{\sqrt{n_0}} W_1^{(0)} x^i + \beta b_1^{(0)}\right)\left(\frac{1}{\sqrt{n_0}} W_1^{(0)} x^j + \beta b_1^{(0)}\right)\right] = \frac{1}{n_0} x^i \cdot x^j + \beta^2.$$

This proves the base case. We will now be working in the setup where

$$f_\theta(x) = \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \sigma\left(\tilde{\alpha}^{(\ell)}(x)\right) + \beta b^{(\ell)} \qquad \left(= \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \alpha^{(\ell)}(x) + \beta b^{(\ell)}\right).$$

For the inductive step, let us assume that for any $n_\ell \in \mathbb{N}$, the n_ℓ random vectors

$$\left(\tilde{\alpha}_{k'}^{(\ell)}(x^1), ..., \tilde{\alpha}_{k'}^{(\ell)}(x^N)\right)^T$$

indexed by $k' = 1, ..., n_\ell$ are i.i.d. Gaussian with zero mean vector and co-variance of the form

$$X := \Sigma_{ij}^{(\ell)} = \mathbb{E}_{(f(x^i), f(x^j)) \sim \mathcal{N}\left(0, \Lambda^{(\ell-1)}(x^i, x^j)\right)}\left[\sigma(f(x^i))\sigma(f(x^j))\right] + \beta^2,$$

where $i, j \in \{1, ..., N\}$. For $k = 1, ..., n_{\ell+1}$, we now consider the random vectors

$$\left(f_{\theta,k}(x^1), ..., f_{\theta,k}(x^N)\right)^T. \tag{10}$$

Conditioned on the pre-activations

$$Z := \left(\tilde{\alpha}^{(\ell)}(x^1), ..., \tilde{\alpha}^{(\ell)}(x^N)\right),$$

the $n_{\ell+1}$ random vectors (10) are i.i.d. Gaussian with zero mean vector and co-variance of the form

$$\tilde{\Sigma}_{ij}^{(\ell+1)} = \mathbb{E}\left[\left(\frac{1}{\sqrt{n_\ell}} W_1^{(\ell)} \alpha^{(\ell)}(x^i) + \beta b_1^{(\ell)}\right)\left(\frac{1}{\sqrt{n_\ell}} W_1^{(\ell)} \alpha^{(\ell)}(x^j) + \beta b_1^{(\ell)}\right) \,\middle|\, Z\right] = \frac{1}{n_\ell} \alpha^{(\ell)}(x^i) \cdot \alpha^{(\ell)}(x^j) + \beta^2.$$

Now, taking $n_\ell \to \infty$ and using the Law of Large Numbers,

$$\tilde{\Sigma}_{ij}^{(\ell+1)} = \frac{1}{n_\ell} \alpha^{(\ell)}(x^i) \cdot \alpha^{(\ell)}(x^j) + \beta^2 \xrightarrow{\text{P}} \mathbb{E}_{(f(x^i), f(x^j)) \sim \mathcal{N}\left(0, \Lambda^{(\ell)}(x^i, x^j)\right)}\left[\sigma(f(x^i))\sigma(f(x^j))\right] + \beta^2 = \Sigma_{ij}^{(\ell+1)} \tag{11}$$

This implies that (10) converges in distribution to a Gaussian r.v. with zero mean vector and co-variance with terms $\Sigma_{ij}^{(\ell+1)}$. Indeed, for any $B \in \mathcal{B}(\mathbb{R}^{n_{\ell+1}})$

$$\mathbb{P}(X \in B) = \int_B \int_{\mathbb{R}^{Nn_l}} \phi_{\left(0, \tilde{\Sigma}^{(\ell+1)}(y)\right)}(x) \phi_{\left(0, \Sigma^{(\ell)}\right)}(y) dy dx,$$

where we let $\phi_{(m,\Sigma)}$ denote a Gaussian p.d.f with mean vector m and co-variance matrix Σ. The Lebesgue Dominated Convergence Theorem and (11) now imply that as $n_\ell \to \infty$, we have

$$\mathbb{P}(X \in B) \to \int_B \int_{\mathbb{R}^{Nn_l}} \phi_{\left(0, \Sigma^{(\ell+1)}\right)}(x) \phi_{\left(0, \Sigma^{(\ell)}\right)}(y) dy dx = \int_B \phi_{\left(0, \Sigma^{(\ell+1)}\right)}(x) dx.$$

□

A.2 Proof of Theorem 2.10

Proof of Theorem 2.10. [4] We will prove the result by induction starting from $L = 1$. In ths case,

$$f_\theta(x) = \frac{1}{n_0} W^{(0)} x + \beta b^{(0)}.$$

It is straightforward to see that for any $k, k' = 1, ..., n_1$, we have

$$\Theta^{(1)}_{kk'}(x, x') = \frac{1}{n_0} \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} x_i x'_i \delta_{jk} \delta_{jk'} + \beta^2 \sum_{j=1}^{n_1} \delta_{jk} \delta_{jk'} = \frac{1}{n_0} x^T x' \delta_{kk'} + \beta^2 \delta_{kk'} = \Sigma^{(1)}(x, x') \delta_{kk'}.$$

This proves the base case. We will now be working in the setup where

$$f_\theta(x) = \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \sigma\left(\tilde{\alpha}^{(\ell)}(x)\right) + \beta b^{(\ell)} \qquad \left(= \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \alpha^{(L)}(x) + \beta b^{(\ell)}\right). \tag{12}$$

For the inductive step, let us assume that as $n_1, ..., n_{\ell-1} \to \infty$ sequentially,

$$\Theta^{(\ell)}_{ii'}(x, x') = \left(\partial_{\tilde{\theta}} \tilde{\alpha}^{(\ell)}_i(x; \theta)\right)^T \partial_{\tilde{\theta}} \tilde{\alpha}^{(\ell)}_{i'}(x'; \theta) \xrightarrow{\text{P}} \Theta^{(\ell)}_\infty(x, x') \delta_{ii'},$$

where we use the notation $\partial_{\tilde{\theta}}$ to denote a vector of partial derivatives with respect to model parameters in the layers $1, ..., \ell$. Applying the chain rule to (12), we have

$$\partial_{\tilde{\theta}} f_{\theta,k}(x) = \frac{1}{\sqrt{n_\ell}} \sum_{i=1}^{n_\ell} \partial_{\tilde{\theta}} \tilde{\alpha}^{(\ell)}_i(x; \theta) \dot{\sigma}(\tilde{\alpha}^{(\ell)}_i(x; \theta)) W^{(\ell)}_{ik}. \tag{13}$$

From (13), it is clear that

$$\left(\partial_{\tilde{\theta}} f_{\theta,k}(x)\right)^T \partial_{\tilde{\theta}} f_{\theta,k'}(x') = \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \sum_{i'=1}^{n_\ell} \Theta^{(\ell)}_{ii'}(x, x') \dot{\sigma}\left(\tilde{\alpha}^{(\ell)}_i(x; \theta)\right) \dot{\sigma}\left(\tilde{\alpha}^{(\ell)}_{i'}(x'; \theta)\right) W^{(\ell)}_{ik} W^{(\ell)}_{i'k'}.$$

Now, we wish to perform two limiting operations. First we take $n_1, ..., n_{\ell-1} \to \infty$ sequentially. Then, by the inductive hypothesis,

$$\left(\partial_{\tilde{\theta}} f_{\theta,k}(x)\right)^T \partial_{\tilde{\theta}} f_{\theta,k'}(x') \xrightarrow{\text{P}} \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \Theta^{(\ell)}_\infty(x, x') \dot{\sigma}\left(\tilde{\alpha}^{(\ell)}_i(x; \theta)\right) \dot{\sigma}\left(\tilde{\alpha}^{(\ell)}_i(x'; \theta)\right) W^{(\ell)}_{ik} W^{(\ell)}_{ik'}.$$

Now, taking $n_\ell \to \infty$ and using the Law of Large Numbers, we have (thanks to Proposition 1.3) that

$$\left(\partial_{\tilde{\theta}} f_{\theta,k}(x)\right)^T \partial_{\tilde{\theta}} f_{\theta,k'}(x') \xrightarrow{\text{P}} \Theta^{(\ell)}_\infty(x, x') \dot{\Sigma}^{(\ell+1)}(x, x') \delta_{kk'}. \tag{14}$$

Finally, we need to consider the contribution of parameters in layer ℓ to our neural tangent kernel

$$\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} \sigma\left(\tilde{\alpha}^{(\ell)}_i(x; \theta)\right) \sigma\left(\tilde{\alpha}^{(\ell)}_{i'}(x'; \theta)\right) \delta_{jk} \delta_{jk'} + \beta^2 \sum_{j=1}^{n_1} \delta_{jk} \delta_{jk'}. \tag{15}$$

Taking $n_\ell \to \infty$ and using the Law of Large Numbers we have (once again thanks to Proposition 1.3) that (15) converges in probability to

$$\Sigma^{(\ell+1)} \delta_{kk'}. \tag{16}$$

Adding the two contributions given by (14) and (16) gives the desired result. □

A.3 Link to kernel methods

While we speak freely of the co-variance kernel without identifying a feature map, one may wonder how the recursion in 1.3 can be written in terms of inner products of feature maps to fit into the general framework of kernels [8]. Indeed, at $\ell = 1$ we have

$$\Sigma^{(1)}(x, x') = \frac{1}{n_0} x^T x' + \beta^2 = \langle \Phi_0(x), \Phi_0(x') \rangle_1$$

where we set

$$\Phi_0(x) = \left(\frac{1}{\sqrt{n_0}} x_1, ..., \frac{1}{\sqrt{n_0}} x_{n_0}, \beta \right)^T$$

and let $\langle \cdot, \cdot \rangle_1$ denote the Euclidean inner product in $\mathbb{R}^{n_0+1}$. Henceforth, we will let $\langle \cdot, \cdot \rangle$ denote the expectation of the Euclidean inner inner product of two random vectors in $\mathbb{R}^2$. For $\ell = 1$, we thus have

$$\Sigma^{(2)}(x, x') = \langle \Phi_1 \left(\Phi_0(x) \right), \Phi_1 \left(\Phi_0(x') \right) \rangle,$$

where we set

$$\Phi_1 \left(\Phi_0(x) \right) = \left(\sigma(f(x)), \beta \right)^T$$

and where we assume that for any $(x, x') \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_0}$, $(f(x), f(x')) \sim \mathcal{N}\left(0, \Lambda^{(1)}(x, x')\right)$. Similarly, for $\ell = 2, ..., L-1$, we have

$$\Sigma^{(\ell+1)}(x, x') = \left\langle \Phi_\ell \left(\left(\sigma(f(x)), \beta \right)^T \right), \Phi_\ell \left(\left(\sigma(f(x)), \beta \right)^T \right) \right\rangle$$

where we set

$$\Phi_\ell \left(\left(\sigma(f(x)), \beta \right)^T \right) = \left(\sigma(f(x)), \beta \right)^T,$$

and where we assume that for any $(x, x') \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_0}$, in the input we have $(f(x), f(x')) \sim \mathcal{N}\left(0, \Lambda^{(\ell-1)}(x, x')\right)$ and in the output we have $(f(x), f(x')) \sim \mathcal{N}\left(0, \Lambda^{(\ell)}(x, x')\right)$. Using this notation, we have

$$\Sigma^{(\ell+1)}(x, x') = \langle \Phi_\ell(....\Phi_1(\Phi_0(x))), \Phi_\ell(....\Phi_1(\Phi_0(x'))) \rangle,$$

for any $\ell = 1, ..., L-1$. This result shows how a co-variance function $\Sigma^{(\ell+1)}$ arising in the $n_1, ..., n_L \to \infty$ limit of our neural network can be viewed as a kernel with a feature map obtained by composing feature maps corresponding to co-variance kernels at inner layers.

References

[1] Arora S, Du SS, Hu W, Li Z, Salakhutdinov R, Wang R. On Exact Computation with an Infinitely Wide Neural Net. *Arxiv.* 2019. Available from https://arxiv.org/abs/1904.11955. [Accessed: 28th November 2019]

[2] Cho Y, Saul LK. Large-margin classification in infinite neural networks. Neural Computation. Volume 22 Issue 10, October 2010. 2678-2697.

[3] Gelfand IM, Fomin SV. *Calculus of Variations.* Revised English Edition. Englewood Cliffs, NJ: Prentice Hall; 1963.

[4] Jacot A, Gabriel F, Hongler C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *Arxiv.* 2018. Available from https://arxiv.org/abs/1806.07572. [Accessed: 28th November 2019]

[5] Lee J, Bahri Y, Novak R, Schoenholz SS, Pennington J, Sohl-Dickstein J. Deep Neural Networks as Gaussian Processes. *Arxiv.* 2018. Available from https://arxiv.org/abs/1711.00165. [Accessed: 3 December 2019]

[6] Neal RM. Bayesian Learning for Neural Networks. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[7] Rudin W. *Real and Complex Analysis.* Mathematics Series. 3rd Edition. New York: McGraw-Hill Book Company; 1987.

[8] Shawe-Taylor J, Cristianini N. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York. 2004.

[9] Schölkopf B, Smola A, Müller KR. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation. 1998. Available from https://www.mlpack.org/papers/kpca.pdf. [Accessed: 6 December 2019]